Tipo de Regresor	Modelo Matemático	Función de Costo	Estrategia de optimización
Linear Regressor	Aproxima una relación entre un conjunto de variables de entrada $X \in \mathbb{R}^{N \times p}$ y una variable objetivo $t \in \mathbb{R}^N$: $y(x,w) = w^T x + w_0$ $y = Xw + b$.	Error cuadrático medio: MSE $J(w) = \frac{1}{2N} \sum_{n=1} (t_n - w^T x_n)^2$	Solución Analítica Cerrada: $\nabla_w J(w) = \frac{-1}{N} X^T (t - Xw) = 0$ Solución: $w = (X^T X) t^T X$ Resuelve el sistema: $X^T X w = X^T t$
Lasso	Introduce una regularización L1, reduciendo la magnitud de los pesos: $y(x) = w^T x + b$.	Minimiza el MSE y mantiene los coeficientes w pequeños: $J(w) = \frac{1}{2N} \sum (t_i - w^T x_i - b)^2 + \lambda \sum \lvert w_j \rvert$ con $\lambda > 0$	Solución numérica Iterativa: → Descenso por coordenadas: Un paso a la vez. → Subgradiente Descendente: → Least Angle Regression: Construye la solución paso a paso a medida que w entra o sale.
ElasticNet	El modelo sigue siendo el mismo $y = w^T x + b$, pero agrega agrega regularización L1 (Lasso) y L2 (Ridge).	$J(w) = \frac{1}{2N} \sum (t_i - w^T x_i - b)^2 + \lambda \left[\alpha \lVert w \rVert_1 + \frac{(1-\alpha)}{2} \lVert w \rVert_2^2 \right]$ $\alpha = 1$ (Lasso), $\alpha = 0$ (Ridge) $0 < \alpha < 1$ (ElasticNet).	Optimización iterativa convexa → Descenso por coordenadas: Parámetro w_j a la vez $w_j \leftarrow \frac{S(z_j, \lambda\alpha)}{1 + \lambda(1-\alpha)}$
Kernel Ridge	Extiende la regresión lineal al espacio de características de un Kernel, se proyectan por una función no lineal: $y(x) = w^T \phi(x)$. $K(x x^T) = \phi(x)^T \phi(x')$	Error cuadrático con L2: $J(w) = \frac{1}{2N} \sum (t_i - w^T \phi(x_i))^2 + \lambda/2 \lVert w \rVert^2$. Derivando w: $w = \phi^T (\phi \phi^T + \lambda I)^{-1} t$.	El problema es convexo y diferenciable, se puede resolver de forma analítica: $\alpha = (K + \lambda I)^{-1} t$ Se usa SVD para invertir $K + \lambda I$ eficientemente.
SGD Regressor = Stochastic Gradient Descent	Mismo modelo de: $y(x) = w^T x + b$.	Diferentes funciones de pérdida según la regresión; más común: squared lost = $J(w) = \frac{1}{2N} \sum (t_i - (w^T x_i + b))^2$ Puede incluir L1 y L2 y elastic Net	No calcula el gradiente usando todos los datos, sino una sola muestra por iteración. La solución es iterativa.

Tipo de Regresor	Modelo Matemático	Función de costo	Estrategia de Optimización.
Bayesian Ridge	Parte del mismo: $y(x) = w^T x + b$. pero no estima un valor para w y b, sino distribuciones para cada uno.	$\log p(w\|t) = -\frac{\beta}{2}\|t - Xw\|^2 - \frac{\alpha}{2}\|w\|^2$ α y β no son fijos, son variables aleatorias.	La inferencia se hace de forma analítica porque las distribuciones son gaussianas
Gaussian Process Regressor	Es una extensión no paramétrica de la regresión bayesiana lineal. $f(x) \sim gp(m(x), k(xx))$.	Se ajustan los hiperparámetros del kernel y del ruido: $\log p(t\|X) = -1/2\, t^T (K + \theta^2 nI)^{-1} t - 1/2 \log [K + \theta^2 nI] - N/2 \log (2\pi)]$.	2 Etapas: a) Inferencia: Distribución posterior: $p(f_* \| X, t, x) = N(f, var(f))$. $f_* = K_T x [K + \theta^2 nI]^{-1} t$. b) Optimización de hiperparámetros mediante gradiente descendente.
Support Vector Machine Regressor	Busca una función plana $f(x)$ que se aleje de los datos en menos de un margen, penalizando errores que se le alejen: $f(x) = w^T \phi(x) + b$.	El problema de optimización es: $\min \frac{1}{2}\|w\|^2 + C \sum (\epsilon_i + \epsilon_i^*)$. $\epsilon \rightarrow$ errores que exceden el margen.	Se usan multiplicadores de Lagrange: $\max -1/2 (\alpha - \alpha^*)^T K (\alpha - \alpha^*) - \varepsilon \sum (\alpha_i + \alpha_i^*) + \sum t_i (\alpha_i - \alpha_i^*)$ Se obtiene la solución usando métodos de optimización cuadrática.
Random forest Regressor	Es un modelo de ensamble formado por B árboles de decisión independiente: $f(x) = \frac{1}{B} \sum_{b=1} f_b(x)$ Cada árbol predice un valor y se toma la media	Cada árbol se entrena minimizando el MSE $= \frac{1}{N} \sum_{i=1} (t_i - y_i)^2$. Se elige la característica y el punto de corte: $split(j, s) = \text{argmin} \cdot [MSE_{izq} + MSE_{der}]$.	Tiene 2 niveles de aleatoriedad controlada: 1. Bootstrap Sampling. 2. Factor Bagging. 3. Crecimiento de árboles. 4. Agregación (media final).

3.

Tipo de Regresor	Modelo Matemático	Función de Costo	Estrategia de Optimización
Gradiente Boosting Regressor	El modelo final es una suma ponderada de árboles débiles: $F(x) = \sum_{m=1}^{M} v \cdot h(x)$. $M \rightarrow$ Número de iteraciones. $h(x)$: Árbol de decisión entrenado. $v \in [0,1]$: learning rate. $F(x)$: predicción final	Trabaja con cualquier función de pérdida diferenciable $L(y, f(x))$. En regresión clásica: $L(y, f(x)) = \frac{1}{2}(y - f(x))^2$. Minimiza la pérdida total sobre todos los datos.	Aplica el descenso del gradiente, pero en el espacio de las funciones, no en el de los parámetros. 1. Función que minimice. 2. Gradientes negativos. 3. Ajuste del árbol al residuo. 4. Peso óptimo del árbol. 5. Actualizar modelo. 6. Repetir M iteraciones.
XG Boost	Es un modelo aditivo como el GBR, pero añade regularización explícita y una aproximación de 2do orden al gradiente. $F(x) = \sum f_m(x)$, $f_m \in F$. $F \rightarrow$ conjunto de árboles. $M \rightarrow$ Número de iteraciones.	En cada iteración, minimiza $\mathcal{L} = \sum_{i=1} L(y_i, \hat{y}_i) + \sum \Omega(f_m)$. $L(y_i, \hat{y}_i)$ es la función de pérdida. $\Omega \rightarrow$ es la penalización sobre la complejidad del árbol.	Utiliza un desarrollo de segundo orden de la pérdida: $L(y_i, \hat{y}_i + f_m(x_i)) \approx L(y_i, \hat{y}_i) + g_i f_m(x_i) + \frac{1}{2} h_i f_m(x_i)^2$. $g \rightarrow$ Gradiente. $h \rightarrow$ hessiano.

Tipo de Regresor	Mínimos Cuadrados	Máxima Verosimilitud	Máximo a Posteriori	Bayesiano lineal	Regresión Rígida Kernel	Procesos Gaussiano's	Escalabilidad
Lineal Regressor	Minimizar el MSE entre predicciones y valores reales: Formulación: $\min J(w) = \frac{1}{2N}\sum_{n=1}^{N}(t_n - w^T x_n - b)^2$ Solución: $w = (X^T X)^{-1} X^T t$ Resuelvo a través de: SVD o QR.	Estimador bajo ruido gaussiano: $t_n = w^T x_n + \epsilon_n$ Formulación: $p(t\|X, w, \theta^2) = \prod N(t_n \| w^T x_n, \theta^2)$ Relación: $\arg\max p(t\|x w) = \arg\min \|t - Xw\|^2$.	Introduce creencia previa. Se convierte en "ridge regressor" agregar regularización. Solución: $w_{MAP} = (X^T X + \lambda I)^{-1} X^T t$.	Considera toda la distribución sobre los pesos Solución: $p(w\|t) = N(w\|M_N, S_N)$. $p(w) = N(0, \alpha^{-1} I)$	Espacio de características no lineal Formulación = $\min_w \frac{1}{2}\|t - \phi w\|^2 + \frac{\lambda}{2}\|w\|^2$. Para lineal regressor: caso Kernel lineal $(K = XX^T)$.	Modelar función como distribución gaussiana: $f(x) \sim GP(0, K(XX^T))$ El lineal regressor usa un GP lineal y sin incertidumbre. 1	Si hay muchas características el costo crece y se deben usar métodos aproximados.
Lasso	$J_{lasso}(w) = \frac{1}{2N}\sum(t - w^T x_i - b)^2 + \lambda\sum\|w_j\|_1$.	La función de costo puede verse como una verosimilitud max regularizada: $\log p(w\|t) = \log p(t\|w) + \log p(w)$. $p(w)$ laplaciano	Introduce un prior laplaciano: $\log p(w\|t) = -\log p(t\|w) - \log p(w)$. $p(w) = \frac{\lambda}{2} e^{-\lambda\|w\|}$	Prior laplaciano: de la forma: $p(w) = N(0, \alpha^{-1} I)$ $p(w) = \alpha \cdot e^{-\lambda\|w\|}$ ya no es una gaussiana cerrada.	Kernel lasso: $\min_w \frac{1}{2}\|t - \phi w\|^2 + \lambda\|w\|_1$	El Prior ya no es gaussiano Si se define con ruido laplaciano, se puede definir como Lasso..	El modelo debe resolverse con métodos iterativos, lo que afecta el tiempo de procesamiento y memoria Lento con P grande.
ElasticNet	El modelo matemático parte de mínimos cuadrados añadiendo L1 + L2	El segundo término actúa como una penalización bayesiana híbrida (Laplaciana + Gaussiana).	$-\log p(t\|w) =$ proviene del ruido gaussiano. $\log p(w) =$ prior mixto $(L_1 + L_2)$.	$p(w) \propto e^{-\lambda(\alpha\|w\|_1 + \frac{(1-\alpha)}{2}\|w\|_2^2)}$	Kernel ElasticNet: $J_{kernel}(w) = \frac{1}{2}\|t - \phi w\|^2 + \lambda[\alpha\|w\|_1 + \frac{(1-\alpha)}{2}\|w\|_2^2]$	Se puede considerar como una mezcla de gaussiano y regularizada.	Tiene mejor estabilidad y convergencia por L2. Pero se puede afectar por α, λ y los datos deben estar normalizados
Kernel Ridge	Versión no lineal y regularizada. Introduce transformaciones no lineales de las variables. $J(w) = 1/2 N \sum (t_i - w^T \phi(x))^2 + \lambda/2 \|w\|_2$	Espacio de características $\phi(x)$ $t_i = w^T \phi(x_i) + \epsilon$. Agrega regularización L2.	$p(t\|w) = N(\phi w, \beta^{-1} I) \Rightarrow p(w) = N(0, \alpha^{-1} I$. El log posterior es: $J(w) = \beta/2 \|t - \phi w\|^2 + \alpha/2 \|w\|^2$. $\lambda = \alpha/\beta \rightarrow$ Función de costo	$p(w\|t) = N(w\|M_N, S_N)$ $S_N^{-1} = \alpha I + \beta x^T x$, $M_N = \beta S_N X^T I$. X se reemplaza por la matriz de características.	Puede operar en espacios no lineales mediante el kernel.	Tiene la misma fórmula de: $p(t\|x) = N(0, K + \theta^2 I)$ pero y es tratado como un valor determinista..	Tiene mayor potencia en modelos no lineales pero su complejidad cúbica limita el uso en grandes volúmenes de datos.

Tipo de Regresor	Mínimos Cuadrados	Max Verosimilitud	Max a posteriori	Bayes lineal	Regresión rígida K.	Procesos Gaussianos	Escalabilidad
SGD Regressor	Minimiza la misma función de costo, pero iterativamente usando gradientes por muestra: $w \leftarrow w - \eta \nabla_w$	Implementa aprox estocástica actualizando para cada muestra de forma progresiva.	Incluye: Prior Gaussiano // laplaciano // mixto Adapta según el regresor	Obtiene solo el valor más probable mediante actualizaciones secuenciales.	Trabaja en el espacio original de las características y no usa Kernels.	Se puede considerar como una versión lineal de GP	Tiene eficiencia media. Permite entrenar modelos lineales sobre millones de muestras y variables.
Bayesian Ridge	Incluye regularización e inferencia probabilística: $J(w) = \frac{\beta}{2}\|t - Xw\|^2 + \frac{\alpha}{2}\|w\|^2$	La inferencia sobre $p(w)$ es distribucional. $p(w\|t,\alpha,\beta) \propto p(t\|X,w,\beta) \cdot p(w\|\alpha)$.	Adapta una distribución gaussiana sobre $p(w)$.	Versión paramétrizada del modelo donde α y β se estiman de los datos directamente.	$J(w) = \beta/2 \|t - Xw\|^2 + \alpha/2 \|w\|^2$. Es un ridge probabilístico que obtiene distribuciones de incertidumbre sobre los pesos.	Se considera un GP con espacio de características finito z	Cumbre precisión probabilística con costo razonable. No es tan escalable como SGD Regressor.
Gaussian Process Regressor	Define una distribución sobre funciones. Entrega una familia infinita de curvas con media y varianza. $f(x) \sim GP(0, K(xx'))$ sin prior ni incertidumbre	No hay parámetros fijos, se maximiza la verosimilitud marginal sobre las funciones: $p(t\|x) = N(0, K + \theta^2 I)$.	Se integra sobre el espacio de funciones completo.	Define una matriz de covarianza $K = xx^T$. Cuantifica la incertidumbre.	Es el bayesian ridge con infinitas bases implícitas.	La media es igual al resultado del Kernel Ridge, pero sin incertidumbre.	Es el más completo de la familia bayesiana, pero es solo bueno para conjuntos medianos. Tiene incertidumbre real.
Support Vector Machine Regressor	Minimiza la norma de los pesos + penaliza solo los errores mayores a ε Ignora errores pequeños	Tiene max verosimilitud con ruido laplaciano truncado. Tiene zona de tolerancia	Usa una función de pérdida lineal por tramos	Busca modelar el ruido con una tolerancia fija	Busca minimizar la pérdida ε. $f(x) = \sum (a_i - a_i^*) K(x_i, x^*) + b$.	Es de naturaleza determinista y no se considera gaussiana, solo trabaja con la media.	La versión lineal se usa para muchos datos, pero Kernel SVR solo para datasets medianos Pobre con N.
Random Forest Regressor	Ajusta múltiples funciones locales, cada una válida en una región del espacio. Puede aproximar regiones no lineales.	No reduce el error Max Verosimilitud, reduce la varianza por ensamble. Se asume media + ruido gaussiano iid.	No es un estimador MAP clásico. Es un ensamble No paramétrico de NSE por nodo.	La incertidumbre que se reporta suele ser empírica, no una ventaja bayesiana bien calibrada.	Se puede ver como una regresión Kernel, pues dos puntos pueden tener similitud si caen en la misma hoja y tener un mismo núcleo.	Comparte la idea de promediar puntos similares, pero no hay prior ni posteriori probabilístico	Es escalable, paralelizable y resistente al ruido, puede estimar incertidumbre por varianza entre árboles.

Tipo de Regresor	Min Cuadrados	Max Verosimilitud	Max a posteriori	Bayes lineal	Ridge Kernel	Procesos Gaussianos	Escalabilidad
Gradient Boosting Regressor	Minimiza $\sum L(y, f(x_i))$ por gradiente funcional y tiene solución iterativa. Tiene regularización implícita con el número de árboles.	Minimiza la pérdida cuadrática, que es equivalente a maximizar la verosimilitud. Realiza descenso del gradiente sobre el log-likehood.	No asume ningún prior ni probabilidad, pero sí regulariza implícitamente. El learning rate actúa como un prior suave.	Ambos son promedios ponderados, pero el GBR es iterativo sobre un modelo flexible.	Tiene linealidad a través de los árboles y suma los árboles que corrigen errores, como los Kernels.	Ambos construyen modelos de funciones no lineales, pero el GBR lo hace determinísticamente. Si el Kernel tiene regiones locales, su medias se comporta similar al ajuste por Boosting.	El costo depende del número de árboles M y de muestras N. No es bueno para $N > M$, pero es robusto y tiene mejor sensibilidad.
XG Boost	Si se usa $l(y, \hat{y}_i) = \frac{1}{2}(y - \hat{y})^2$ y $\lambda = 0$, es un GBR, pero con regularización actúa como un Ridge No lineal	Minimiza pérdidas equivalentes a log-likelihood negativos, pero la regularización explícita Ω convierte el entrenamiento en un MAP. Con un modelo no lineal.	Es una implementación determinista del MAP, donde el prior está codificado en los hiperparámetros $(\lambda, \gamma, \alpha)$.	Cada hoja tiene una creencia previa de que los valores deben ser pequeños.	Es una versión discretizada, jerárquica y regularizada del Kernel, donde los árboles reemplazan los Kernels.	No usa Kernels explícitos, pero cada árbol tiene regiones locales donde se agrupan puntos similares. No tiene varianzas ni posterior gaussiano.	Es eficiente y robusto, logra suavidad local y escalabilidad masiva.